

270017 - PROP - Proyectos de Programación

Unidad responsable: 270 - FIB - Facultad de Informática de Barcelona
Unidad que imparte: 723 - CS - Departamento de Ciencias de la Computación
Curso: 2015
Titulación: GRADO EN INGENIERÍA INFORMÁTICA (Plan 2010). (Unidad docente Obligatoria)
Créditos ECTS: 6 Idiomas docencia: Catalán, Castellano

Profesorado

Responsable: - Jordi Delgado Pin (jdelgado@cs.upc.edu)
Otros: - Alicia Maria Ageno Pulido (ageno@cs.upc.edu)
- Borja Valles Fuente (valles@cs.upc.edu)
- Horacio Rodríguez Hontoria (horacio@cs.upc.edu)
- Jordi Turmo Borrás (turmo@cs.upc.edu)
- Miquel Sanchez Marre (miquel@cs.upc.edu)

Capacidades previas

- Capacidad para resolver problemas algorítmicos de dificultad media a partir de una especificación clara, y de implementar las soluciones en un lenguaje de programación imperativo.
- Conocimiento de los mecanismos básicos de estructuración de programas (modularización, encapsulación, tipos abstractos de datos, clases) y capacidad para aplicarlos a problemas pequeños-medios (unos pocos módulos)
- Conocimiento de los elementos de programación orientada a objetos (clases, objetos, mecanismos de ejecución).
- Familiaridad con un lenguaje imperativo orientado a objetos.
- Capacidad de usar y programar estructuras de datos en este lenguaje.
- Capacidad de usar librerías en este lenguaje.
- Dominio de estrategias básicas para encontrar y corregir errores en módulos sencillos.

Requisitos

- Corequisito IES
- Pre-requisito EDA
- Pre-Corequisito BD

Competencias de la titulación a las cuales contribuye la asignatura

Específicas:

- CT2.3. Diseñar, desarrollar, seleccionar y evaluar aplicaciones, sistemas y servicios informáticos, y al mismo tiempo asegurar su fiabilidad, su seguridad y su calidad, conforme a principios éticos y a la legislación y la normativa vigente.
- CT2.4. Demostrar conocimiento y capacidad de aplicación de las herramientas necesarias para el almacenaje, el procesamiento y el acceso a los Sistemas de información, incluidos los basados en web.
- CT2.5. Diseñar y evaluar interfaces persona-computador que garanticen la accesibilidad y la usabilidad a los sistemas, a los servicios y a las aplicaciones informáticas.
- CT3.6. Demostrar conocimiento de la dimensión ética en la empresa: la responsabilidad social y corporativa en

270017 - PROP - Proyectos de Programación

general y, en particular, las responsabilidades civiles y profesionales del ingeniero en informática.

CT4.1. Identificar las soluciones algorítmicas más adecuadas para resolver problemas de dificultad mediana.

CT4.2. Razonar sobre la corrección y la eficiencia de una solución algorítmica.

CT4.3. Demostrar conocimiento y capacidad de aplicación de los principios fundamentales y las técnicas básicas de los sistemas inteligentes y su aplicación práctica.

CT5.1. Escoger, combinar y explotar diferentes paradigmas de programación, en el momento de construir software, atendiendo a criterios como la facilidad de desarrollo, la eficiencia, la portabilidad y la mantenibilidad.

CT5.2. Conocer, diseñar y utilizar de forma eficiente los tipos y las estructuras de datos más adecuados para la resolución de un problema.

CT5.3. Diseñar, escribir, probar, depurar, documentar y mantener código en un lenguaje de alto nivel para resolver problemas de programación aplicando esquemas algorítmicos y usando estructuras de datos.

CT5.4. Diseñar la arquitectura de los programas utilizando técnicas de orientación a objetos, de modularización y de especificación e implementación de tipos abstractos de datos.

CT5.5. Usar las herramientas de un entorno de desarrollo de software para crear y desarrollar aplicaciones.

CT5.6. Demostrar conocimiento y capacidad de aplicación de los principios fundamentales y de las técnicas básicas de la programación paralela, concurrente, distribuida y de tiempo real.

CT6.1. Demostrar conocimiento y tener capacidad para administrar y mantener sistemas, servicios y aplicaciones informáticas.

CT8.6. Demostrar comprensión de la importancia de la negociación, de los hábitos de trabajo efectivos, del liderazgo y de las habilidades de comunicación en todos los entornos de desarrollo de software.

CT8.7. Controlar versiones y configuraciones del proyecto.

Genéricas:

G1. EMPRENDEDURÍA E INNOVACIÓN: Conocer y comprender la organización de una empresa y las ciencias que rigen su actividad; capacidad de comprender las reglas laborales y las relaciones entre la planificación, las estrategias industriales y comerciales, la calidad y el beneficio. Desarrollar la creatividad, el espíritu emprendedor y la tendencia a la innovación.

G5. TRABAJO EN EQUIPO: Ser capaz de trabajar como miembro de un equipo, ya sea como un miembro más, o realizando tareas de dirección con la finalidad de contribuir a desarrollar proyectos con pragmatismo y sentido de la responsabilidad, asumiendo compromisos teniendo en cuenta los recursos disponibles.

270017 - PROP - Proyectos de Programación

Metodologías docentes

El temario se expone de forma muy práctica, a través de la presentación de muchos ejemplos.

Las clases de teoría introducen todo los conocimientos, las técnicas y los conceptos necesarios para la realización del proyecto de la asignatura. Algunas de las clases de laboratorio, normalmente hacia principio de curso, sirven para que el profesor presente brevemente las notaciones, lenguajes, librerías y herramientas a utilizar. En la mayoría de clases de laboratorio, sin embargo, no es el profesor quien lleva la iniciativa. Son unas horas que cada grupo de proyecto puede utilizar para trabajar en común, o bien para consultar dudas con el profesor y recibir sus comentarios.

Las dos horas de clases de teoría y las dos de laboratorio se hacen semanalmente, excepto las dos primeras semanas que sólo se hace teoría y las dos últimas que sólo se hace laboratorio.

La realización del proyecto es la parte principal del trabajo a realizar dentro de la asignatura. Este proyecto tendrá un fuerte componente algorítmico y de estructuras de datos.

Para implementar este proyecto, el sistema de trabajo será el siguiente:

- Los estudiantes se organizan en grupos de 4 personas, y los grupos se estructuran en macrogrupos de 3 grupos (los "clusters") de 12 personas.
- Cada "cluster" tendrá un profesor tutor asignado.
- Habrá 3 enunciados diferentes cada cuatrimestre que deberán ser resueltos todos ellos por cada uno de los "clusters" (un enunciado para cada grupo). Aunque los enunciados son diferentes, presentarán suficientes similitudes como para que tenga sentido identificar partes comunes que puedan ser desarrolladas una sola vez y reutilizadas por todo el "cluster".
- Dentro de cada grupo, un estudiante y sólo uno es responsable de programar y probar cada clase.

El curso utiliza el lenguaje de programación Java.

Objetivos de aprendizaje de la asignatura

1. Invocar los conocimientos de programación adquiridos en asignaturas anteriores durante el desarrollo de un proyecto de programación de tamaño medio, eligiendo algoritmos y estructuras de datos adecuados para construir un programa correcto y eficiente.
2. Organizar el trabajo de diseño y programación de un grupo de personas de tamaño medio (de tres a diez), de acuerdo con una planificación temporal preestablecida. Esta organización incluye tanto el cumplimiento de la planificación dada como el reparto de tareas entre los miembros del grupo.
3. Identificar partes susceptibles de ser factorizadas en la especificación, diseño e implementación de un programa, para resolverlas una sola vez. Usar de manera efectiva los mecanismos de la orientación a objetos para hacer la factorización.
4. Conocer los principios del diseño y la programación orientadas a objetos, su justificación, y las ventajas e inconvenientes de adoptar este paradigma en un proyecto de programación.
5. Usar el mecanismo de clases y herencia para evitar la redundancia y para capturar abstracciones de especificación, diseño e implementación. Entender las ventajas de este uso de la orientación a objetos con respecto a fiabilidad, modificabilidad, portabilidad, reusabilidad y productividad.
6. Tratar correctamente las situaciones anómalas dentro del programa, usando, si es conveniente, el mecanismo de excepciones ofrecido por el lenguaje de programación.
7. Diseñar una interfaz de usuario razonablemente cómoda y eficaz para el programa, usando una librería GUI.
8. Desarrollar un proyecto de programación de tamaño medio, en grupo, de acuerdo con un plan de desarrollo y una arquitectura proporcionados de antemano.

270017 - PROP - Proyectos de Programación

Horas totales de dedicación del estudiantado

Dedicación total: 150h	Grupo grande/Teoría:	30h	20.00%
	Grupo mediano/Prácticas:	0h	0.00%
	Grupo pequeño/Laboratorio:	30h	20.00%
	Actividades dirigidas:	6h	4.00%
	Aprendizaje autónomo:	84h	56.00%

270017 - PROP - Proyectos de Programación

Contenidos

Conceptos de OO.

Competencias de la titulación a las que contribuye el contenido:

Descripción:

Especificación, diseño y programación orientadas a objetos (OO). Clases, objetos, atributos, métodos y relaciones. Encapsulamiento y ocultación, abstracción y clasificación, herencia. Polimorfismo, "binding" estático y dinámico. Otras relaciones: asociación, agregación, dependencia.

Relación de la OO con los criterios de calidad del software.

UML: Nomenclatura de clases, atributos y operaciones, clases abstractas, genéricas y herencia. Nomenclatura de los diferentes tipos de relaciones.

Especificación: diagramas de clases y casos de uso.

Competencias de la titulación a las que contribuye el contenido:

Descripción:

Ciclos de vida.

Definición del proyecto. Requerimientos. Especificación funcional. Modelar primero el dominio.

UML: diagrama de clases (dominio). Diagrama de casos de uso. Explicaciones textuales. Estudio de un caso.

Diseño: arquitectura en 3 capas

Competencias de la titulación a las que contribuye el contenido:

Descripción:

Descripción del paradigma de diseño de arquitectura en 3 capas.

El lenguaje de programación Java

Competencias de la titulación a las que contribuye el contenido:

Descripción:

Implementación de los conceptos de OO en Java. Herencia, polimorfismo y genericidad como herramientas de programación. Mecanismos de ejecución. Creación y destrucción de objetos. Llamadas a métodos y paso de parámetros. "Binding" estático y dinámico. "Garbage collection". Implementación de la arquitectura en 3 capas en Java. Implementación de interfaces en Java. Conceptos adicionales: ficheros, serialitzacions, bases de datos.

Depuración de programas

Competencias de la titulación a las que contribuye el contenido:

270017 - PROP - Proyectos de Programación

Descripción:

Verificación. Program Testing: prueba y depuración.
Prueba de componentes.
Estrategias de prueba e integración. Planificación de pruebas.
"Andamiaje" y tests de regresión.
Herramientas de ayuda a la prueba.
Tratamiento de errores y excepciones.

Documentación

Competencias de la titulación a las que contribuye el contenido:

Descripción:

Documentación técnica y de usuario. Estándares de documentación. Herramientas de ayuda a la documentación (Doxygen).

Conceptos básicos de diseño de interfaces

Competencias de la titulación a las que contribuye el contenido:

Descripción:

Tipos de entrada y salida.
Principios de comunicación con interlocutores humanos. Presentación de información, entrada y validación de datos, navegación, niveles de ayuda, modelos de usuario.

270017 - PROP - Proyectos de Programación

Planificación de actividades

Descripción del proyecto y formación de grupos	Dedicación: 3h Grupo grande/Teoría: 1h Grupo mediano/Prácticas: 0h Grupo pequeño/Laboratorio: 2h Actividades dirigidas: 0h Aprendizaje autónomo: 0h
Descripción: Descripción de las entregas del proyecto. Formación de los grupos de prácticas. Objetivos específicos: 2, 8	
Conceptos básicos de OO	Dedicación: 6h Grupo grande/Teoría: 3h Grupo mediano/Prácticas: 0h Grupo pequeño/Laboratorio: 0h Actividades dirigidas: 0h Aprendizaje autónomo: 3h
Descripción: Desarrollo del tema 1 de la asignatura. Objetivos específicos: 4, 5, 8	
Ampliación conceptos de OO	Dedicación: 4h Grupo grande/Teoría: 2h Grupo mediano/Prácticas: 0h Grupo pequeño/Laboratorio: 0h Actividades dirigidas: 0h Aprendizaje autónomo: 2h
Descripción: Desarrollo del tema 1 de la asignatura. Objetivos específicos: 4, 5, 8	
Especificación: conceptos básicos	Dedicación: 4h Grupo grande/Teoría: 2h Grupo mediano/Prácticas: 0h Grupo pequeño/Laboratorio: 0h Actividades dirigidas: 0h Aprendizaje autónomo: 2h

270017 - PROP - Proyectos de Programación

Descripción:

Desarrollo del tema 2 de la asignatura. Aplicación de los conceptos introducidos al proyecto.

Objetivos específicos:

3, 5, 8

Ejemplo de un caso práctico

Dedicación: 4h

Grupo grande/Teoría: 2h

Grupo mediano/Prácticas: 0h

Grupo pequeño/Laboratorio: 0h

Actividades dirigidas: 0h

Aprendizaje autónomo: 2h

Descripción:

Desarrollo de un caso práctico, con los conceptos de los temas 1 y 2 de la asignatura.

Objetivos específicos:

3, 4, 5, 8

Java: implementación de los conceptos de OO

Dedicación: 4h

Grupo grande/Teoría: 2h

Grupo mediano/Prácticas: 0h

Grupo pequeño/Laboratorio: 0h

Actividades dirigidas: 0h

Aprendizaje autónomo: 2h

Descripción:

Desarrollo del tema 4 de la asignatura.

Objetivos específicos:

5, 8

Diseño: arquitectura en tres capas

Dedicación: 4h

Grupo grande/Teoría: 2h

Grupo mediano/Prácticas: 0h

Grupo pequeño/Laboratorio: 0h

Actividades dirigidas: 0h

Aprendizaje autónomo: 2h

Descripción:

Desarrollo del tema 3 de la asignatura.

Objetivos específicos:

3, 4, 5, 8

270017 - PROP - Proyectos de Programación

Primera entrega del proyecto	Dedicación: 0h Actividades dirigidas: 0h Aprendizaje autónomo: 0h
Descripción: Especificación y análisis. Se entregará una descripción detallada de las funcionalidades a desarrollar, los casos de uso y un modelo del dominio del problema. Objetivos específicos: 3, 4, 5, 8	
Java: implementación de la arquitectura en tres capas	Dedicación: 4h Grupo grande/Teoría: 2h Grupo mediano/Prácticas: 0h Grupo pequeño/Laboratorio: 0h Actividades dirigidas: 0h Aprendizaje autónomo: 2h
Descripción: Desarrollo del tema 4 de la asignatura. Objetivos específicos: 5, 8	
Documentación	Dedicación: 4h Grupo grande/Teoría: 2h Grupo mediano/Prácticas: 0h Grupo pequeño/Laboratorio: 0h Actividades dirigidas: 0h Aprendizaje autónomo: 2h
Descripción: Desarrollo del tema 6 de la asignatura. Objetivos específicos: 2, 8	
Depuración de programas	Dedicación: 4h Grupo grande/Teoría: 2h Grupo mediano/Prácticas: 0h Grupo pequeño/Laboratorio: 0h Actividades dirigidas: 0h Aprendizaje autónomo: 2h
Descripción: Desarrollo del tema 5 de la asignatura.	

270017 - PROP - Proyectos de Programación

Objetivos específicos: 6, 8	
Especificación de las clases compartidas	Dedicación: 0h Actividades dirigidas: 0h Aprendizaje autónomo: 0h
Descripción: Especificación de las clases compartidas Objetivos específicos: 1, 3, 4, 5, 8	
Java: implementación de interfaces	Dedicación: 4h Grupo grande/Teoría: 2h Grupo mediano/Prácticas: 0h Grupo pequeño/Laboratorio: 0h Actividades dirigidas: 0h Aprendizaje autónomo: 2h
Descripción: Desarrollo del tema 4 de la asignatura. Objetivos específicos: 7, 8	
Conceptos básicos de diseño de interfaces	Dedicación: 4h Grupo grande/Teoría: 2h Grupo mediano/Prácticas: 0h Grupo pequeño/Laboratorio: 0h Actividades dirigidas: 0h Aprendizaje autónomo: 2h
Descripción: Desarrollo del tema 7 de la asignatura. Objetivos específicos: 7, 8	
Java: conceptos adicionales	Dedicación: 4h Grupo grande/Teoría: 2h Grupo mediano/Prácticas: 0h Grupo pequeño/Laboratorio: 0h Actividades dirigidas: 0h Aprendizaje autónomo: 2h

270017 - PROP - Proyectos de Programación

Descripción:

Desarrollo del tema 4 de la asignatura.

Objetivos específicos:

5, 8

Segunda entrega del proyecto

Dedicación: 0h

Actividades dirigidas: 0h

Aprendizaje autónomo: 0h

Descripción:

Diseño y principio de la implementación. Esta entrega incluye el diseño de la posible parte compartida entre los grupos del mismo "cluster" y su implementación. Se detallará el reparto de responsabilidades entre los grupos. Respecto al trabajo propio del grupo, se debe implementar completamente el modelo del dominio y el núcleo básico algorítmico del proyecto, proporcionando los juegos de pruebas de la parte implementada y un diseño preliminar de las pantallas que componen la interfaz de la aplicación .

Objetivos específicos:

1, 2, 3, 4, 5, 6, 8

Conceptos relacionados con el proyecto

Dedicación: 6h

Grupo grande/Teoría: 4h

Grupo mediano/Prácticas: 0h

Grupo pequeño/Laboratorio: 2h

Actividades dirigidas: 0h

Aprendizaje autónomo: 0h

Descripción:

Conceptos relacionados con el proyecto

Examen Teoría

Dedicación: 17h

Actividades dirigidas: 2h

Aprendizaje autónomo: 15h

Descripción:

Este examen evaluará los contenidos introducidos en las clases de teoría.

Objetivos específicos:

1, 4, 5

Tercera entrega del proyecto

Dedicación: 2h

Actividades dirigidas: 2h

Aprendizaje autónomo: 0h

270017 - PROP - Proyectos de Programación

Descripción:

Entrega final. La documentación a presentar debe incluir el diagrama de clases completo (UML) y la implementación completa del proyecto, así como los juegos de prueba (si procede) y el manual de usuario. Además de la entrega de la documentación, cada grupo deberá hacer una presentación de su proyecto.

Objetivos específicos:

1, 2, 3, 4, 5, 6, 7, 8

Supervisión primera entrega

Dedicación: 13h 30m

Grupo grande/Teoría: 0h

Grupo mediano/Prácticas: 0h

Grupo pequeño/Laboratorio: 4h

Actividades dirigidas: 0h

Aprendizaje autónomo: 9h 30m

Descripción:

Clases de laboratorio relacionadas con la primera entrega.

Objetivos específicos:

1, 2, 3, 4, 5, 8

Supervisión segunda entrega

Dedicación: 30h

Grupo grande/Teoría: 0h

Grupo mediano/Prácticas: 0h

Grupo pequeño/Laboratorio: 10h

Actividades dirigidas: 0h

Aprendizaje autónomo: 20h

Descripción:

Clases de laboratorio relacionadas con la segunda entrega.

Objetivos específicos:

1, 2, 3, 4, 5, 6, 8

Supervisión tercera entrega

Dedicación: 40h

Grupo grande/Teoría: 0h

Grupo mediano/Prácticas: 0h

Grupo pequeño/Laboratorio: 12h

Actividades dirigidas: 0h

Aprendizaje autónomo: 28h

Descripción:

Clases de laboratorio relacionadas con la tercera entrega.

Objetivos específicos:

1, 2, 3, 4, 5, 6, 7, 8

270017 - PROP - Proyectos de Programación

Presentación tercera entrega	Dedicación: 7h 30m Grupo grande/Teoría: 0h Grupo mediano/Prácticas: 0h Grupo pequeño/Laboratorio: 0h Actividades dirigidas: 7h 30m Aprendizaje autónomo: 0h
Descripción: Presentación y entrevista presencial de la tercera entrega. Objetivos específicos: 8	

Sistema de calificación

Nota Final: $0.20 * \text{Nota_Examen_Teoria} + 0.80 * \text{Nota_Projecte}$

Nota_Projecte: $(0.10 * \text{Entrega1} + 0.30 * \text{Entrega2} + 0.60 * \text{Entrega3}) * \text{FT}$

Dado que el proyecto es un trabajo en equipo en el que deben participar todos y sólo los miembros del grupo, la nota final del proyecto está ponderada por FT. Este Factor de Trabajo (FT) es una ponderación ($0 \leq \text{FT} \leq 1$) individual que valora el grado de trabajo de cada miembro del grupo. Se hará público junto con las notas de la tercera entrega, y se determinará a partir del reparto de tareas dentro del grupo (entregado a la Entrega3) y de las presentaciones finales del grupo con su tutor.

Competencia transversal "Trabajo en equipo": Se evalúa usando una rúbrica simple en que el tutor de cada grupo puntúa los diferentes aspectos relacionados con la competencia de cada miembro de los grupos.

Competencia transversal "Emprendeduría e innovación": Se evalúa usando una rúbrica simple en que el tutor de cada grupo puntúa los diferentes aspectos relacionados con la competencia de cada miembro de los grupos.

270017 - PROP - Proyectos de Programación

Bibliografía

Básica:

- Meyer, B. Construcción de software orientado a objetos. 2a ed. Prentice Hall, 1998. ISBN 8483220407.
- Budd, T. An introduction to object-oriented programming. 3rd ed. Addison-Wesley, 2002. ISBN 0201760312.
- Binder, R.V. Testing object-oriented systems: models, patterns and tools. Addison-Wesley, 1999. ISBN 9780321700674.
- Arnold, K.; Gosling, J.; Holmes, D. The Java programming language. 4th ed. Addison-Wesley, 2006. ISBN 0321349806.
- Peralta, A.J.; Rodríguez, H. Ingeniería del software: programación orientada a objetos. Edicions UPC, 1994. ISBN 8476533748.

Complementaria:

- Meyer, B. Object-oriented software construction. 2nd ed. Prentice Hall, 1997. ISBN 0136291554.
- Booch, G.; Rumbaugh, J.; Jacobson, I. The unified modeling language user guide. 2nd ed. Addison-Wesley, 2005. ISBN 0321267974.
- Booch, G.; Rumbaugh, J.; Jacobson, I. El lenguaje unificado de modelado: guía del usuario. 2a ed. Addison-Wesley, 2006. ISBN 9788478290765.
- Rumbaugh, J.; Jacobson, I.; Booch, G. The unified modeling language reference manual. 2nd ed. Addison-Wesley, 2005. ISBN 0321245628.
- Rumbaugh, J.; Jacobson, I.; Booch, G. El Lenguaje unificado de modelado: manual de referencia: UML. 2a ed. Addison-Wesley, 2007. ISBN 9788478290871.
- Lee, R.C.; Teppenhardt, W.M. UML and C++: a practical guide to object-oriented development. 2nd ed. Prentice Hall, 2001. ISBN 0130290408.
- Pressman, R.S.; Maxim, B.R. Software engineering: a practitioner's approach. 8th ed. McGraw Hill Higher Education, 2015. ISBN 9780078022128.
- Jeffries, R.; Anderson, A.; Hendrickson, C. Extreme programming installed. Addison-Wesley, 2001. ISBN 0201708426.
- Cooper, A.; Reimann, R.; Cronin, D.; Noessel, C. About face: the essentials of interaction design. 4th ed. Wiley, 2014. ISBN 9781118766576.

Otros recursos:

Enlace web

<http://java.sun.com/>

<http://mat21.etsii.upm.es/ayudainf/aprendainf/programacion.htm>

<http://www-306.ibm.com/software/rational/>

<http://argouml.tigris.org/>